

Amendments to the Claims

Please amend Claims 1 through 15. Please add new Claims 16 through 20. The Claim Listing below will replace all prior versions of the claims in the application:

Claim Listing:

1. (Currently amended) A Computer computer apparatus for managing and sharing engineering data for chemical engineering processes and plants, comprising:
 - an editor defining (i) class views and (ii) a composite class view of the defined class views, given one or more software applications of interest and each given software application having a respective data model or data view, for each said given software application, the editor providing a class view of the ~~application~~ respective data model;
 - the editor consolidating said class views to form a composite class view ~~formed by a consolidation of the class views;~~ and
 - a data server instantiating a multi-tier data model, there being a core conceptual data model having a plurality of routes between attributes in the composite class view and attributes in the core conceptual data model, ~~wherein~~ the class views ~~[[are]]~~ effectively being one tier of the multi-tier data model, the composite class view~~[[s]]~~ ~~[[are]]~~ effectively being a second tier of the multi-tier data model and the core conceptual data model ~~[[is]]~~ effectively being a third tier, ~~such that a~~ the multi-tier data model having ~~with~~ links between corresponding attributes across ~~[[the]]~~ tiers ~~results,~~ the multi-tier data model enabling providing management and sharing of engineering data of the given software applications with other process and plant engineering applications, ~~such that~~ and enhancing process engineering and plant operations ~~are enhanced.~~
2. (Currently amended) The computer apparatus Apparatus as claimed in Claim 1 further comprising an amalgamator ~~[[for]]~~ that synthesizing synthesizes the class views, the composite class view ~~views~~ and the core conceptual data model into a consolidated multi-tier data model.

3. (Currently amended) The computer apparatus Apparatus as claimed in Claim 1 further comprising a mapper ~~[[for]]~~ that linking links the core conceptual data model attributes to the composite class view and the composite class view attributes to ~~application~~ class views, ~~such that there is~~ and provides a one-to-one mapping between an attribute in the composite class view and a route in the core conceptual data model to corresponding given software applications from which the attribute in the composite class view originated.
4. (Currently amended) The computer apparatus Apparatus as claimed in Claim 3 wherein each class view is represented in terms from the respective given software application, ~~such that and an end user of~~ said given software application is able to access data from the core conceptual data model.
5. (Currently amended) The computer apparatus Apparatus as claimed in Claim 1 wherein the class views, the composite class views view and the core conceptual data model are represented by object oriented programming elements.
6. (Currently amended) The computer apparatus Apparatus as claimed in Claim 5 wherein certain object oriented programming elements are defined by classes; and wherein
the apparatus ~~further comprising a class library editing subsystem for enabling editor~~
enables user creation and editing of definitions of classes.
7. (Currently amended) The computer apparatus Apparatus as claimed in Claim 6 wherein the ~~class library editing subsystem~~ editor employs ~~[[XML]]~~ an Extensible Markup Language.
8. (Currently amended) A method of data modeling, comprising the computer implemented steps of:
 - (a) forming a multi-tier data model with links between corresponding attributes across tiers, a first tier being formed by:

for each of multiple given software applications of interest and having a respective data model, providing a practitioner's view of the given software application using a respective class view of the ~~application~~ respective data model;

a second tier being formed by consolidating class views into a composite class view; and

a third tier being formed by forming a core conceptual data model having a plurality of routes between attributes in the composite class view and attributes in the core conceptual data model; and

(b) ~~using sharing, via~~ the multi-tier data model, ~~enabling sharing of~~ engineering data of the given software applications with other process and plant engineering routines.

9. (Currently Amended) [[A]] The method as claimed in Claim 8 wherein the second tier is ~~further~~ formed by synthesizing the class views into the composite class view.
10. (Currently amended) [[A]] The method as claimed in Claim 8 wherein the step of forming a multi-tier data model further ~~includes~~ comprises producing a one-to-one mapping between an attribute in each ~~application~~ class view to the composite class view, and a one-to-one mapping between an attribute in the composite class view and a route in the core conceptual data model to corresponding given software applications from which the attribute in the composite class view originated.
11. (Currently amended) [[A]] The method as claimed in Claim 8 wherein the step of providing a practitioner's view includes, in each class view, representing the ~~application~~ respective data model in terms from the respective given software application.
12. (Currently amended) [[A]] The method as claimed in Claim 8 further comprising the step of representing at least one of the class views, the composite class view and the core conceptual data model in terms of object oriented programming elements.

13. (Currently amended) [[A]] The method as claimed in Claim 12 wherein certain object oriented programming elements are defined by classes; and
the method further comprising the step of enabling user creation and edition of definitions of classes.
14. (Currently amended) [[A]] The method as claimed in Claim 13 wherein the step of enabling user creation and edition includes employing [[XML]] Extensible Markup Language interfaces.
15. (Currently amended) A computer program product comprising:
(a) a computer readable usable medium [[for]] that manages managing engineering data; and
(b) a set of computer program instructions ~~embodied~~ encoded on the computer readable usable medium, the set of computer program instructions when executed on a computer causing the computer including instructions to:
provide a respective class view for each of plural given software applications of interest and having a respective data model, each class view being of the respective ~~application~~ data model;
form a composite class view from the class views;
form a conceptual model; [[and]]
form a consolidated multi-tier data model from the ~~application~~ class views, the composite class view and the conceptual model; and
via, the consolidated multi-tier data model, provide such that sharing of engineering data of the given software applications ~~is-enabled~~ with other process and plant engineering applications.
16. (New) The computer program product of Claim 15, wherein the consolidated multi-tier data model insulates the given software applications from changes in the conceptual model.

17. (New) The computer program product of Claim 15, wherein the consolidated multi-tier data model is insulated from changes in the given software applications.
18. (New) The computer program product of Claim 15, wherein the consolidated multi-tier data model provides an application independent and normalized data model where the composite class view is application independent.
19. (New) The computer program product of Claim 15, wherein the consolidated multi-tier data model comprises an editor and a class store, the class store providing an interface to the respective class views, the composite class view, and the conceptual model to share data between the consolidated multi-tier data model and the given software applications.
20. (New) The computer program product of Claim 19, wherein the editor and the class store use an Extensible Markup Language, and wherein the composite class view provides for the class views remaining in the given software application domain terminology.